Remarks

The present amendment responds to the Official Action dated January 24, 2006. The Official Action rejected claims 1 and 12 under 35 U.S.C. § 102(b) based on Kitamura et al. U.S. Patent No. 5,197,145 (Kitamura). The Official Action rejected claims 1-9, 12 and 42 under 35 U.S.C. § 103(a) based on Hirata et al. U.S. Patent No. 5,430,851 (Hirata) in view of Blackmon et al. U.S. Patent No. 6,895,482 (Blackmon). The Official Action rejected claims 13-15 under 35 U.S.C. § 103(a) based on Hirata in view of Blackmon and further in view of Kishida et al. U.S. Patent No. 6,065,112 (Kishida). The Official Action also rejected claims 25, 26, and 29 under 35 U.S.C. § 103(a) based on Hirata in view of Cray U.S. Patent No. 3,833,889. The Official Action also rejected claims 30 and 34-36 under 35 U.S.C. § 103(a) based on Hirata in view of Cray and further in view of Ganapathy et al. U.S. Patent No. 6,557,096 (Ganapathy). These grounds of rejection are addressed below. The Official Action also objected to claims 10, 11, 16-24, 27, 28, 31-33, and 37-41 as being dependent upon a rejected base claim, but indicated that they would be allowable if rewritten in independent form. Claim 5 and 25 have been canceled without prejudice. Claim 27 has been rewritten in independent form. Claims 1, 2-4, 6-12, 26, 29-36, and 40-42 have been amended to be more clear and distinct. New claims 43 and 44 have been added. Claims 1-4, 6-24, and 26-44 are presently pending.

The Art Rejections

As addressed in greater detail below, Kitamura, Hirata, Blackmon, Kishida, Cray, and Ganapathy do not support the Official Action's reading of them and the rejections based

thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of Kitamura, Hirata, Blackmon, Kishida, Cray, and Ganapathy made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

Claims 1 and 12 were rejected under 35 U.S.C. § 102(b) based on Kitamura. Kitamura discloses a buffer storage system that is interposed between an instruction executing portion and a main storage. The buffer storage system allows for fast fetching of instructions and data. The identity of the instructions and data are maintained between the contents of the buffer storage system and the corresponding portion of main storage. Kitamura, col. 1, lines 9-17. Kitamura does not describe a method to transform a program by splitting the program into a set of control instructions and a set of arithmetic/logic instructions, reducing the set of AL instructions to a reduced set of AL instructions, assigning to each AL instruction an address in at least one AL memory, and generating instruction fetch (IF) instructions in a sequencing order specified by the IF instructions for programmably selecting AL instructions to be fetched from said at least one AL memory where the IF instructions have an IF instruction format which identifies at least one assigned address of at least one AL instruction as is presently claimed in amended claim 1.

In contrast to Kitamura's buffer storage system, the present invention describes a method to transform a program beginning with an originally compiled or assembled program comprising control structures made up of branches, calls, returns, etc. and non-control execution functions. As part of the present method, the program is split into two parts. A first part is a set of control structure instructions and a second part is the non-control execution functions, such as arithmetic/logic (AL) instructions. The non-control instructions, such as the Al instructions, are

reduced in a list creation function by removing duplicate AL instructions. See, for example, Fig. 2, page 7, line 6 – page 8, line 8, page 11, line 15 - page 12, line 13, and page 30, lines 12-15 of the present invention. Each AL instruction is assigned an address from at least one AL memory. Also, the present invention describes generating instruction fetch (IF) instructions. For example, an instruction addressing control program 226 is generated by the code splitting tool 214 of Fig. 2. The instruction addressing control program 226 is composed of instruction fetch (IF) instructions which are stored in a wings fetch instruction memory (WIM) 312 of Fig. 3. Unique IF instructions are generated with an IF instruction format that is used to identify at least one assigned address of at least one AL instruction. The reduced set of AL instructions are stored in AL memories, which are also referenced as IMemories. See, for example, Figs. 2 and 3, page 7, line 6 – page 8, line 8, page 12, lines 5-23, and page 13, line 20 - page 14, line 8 of the present invention.

With regard to the amended claim 12, Kitamura does not describe a code splitting tool for generating an instruction addressing control program as a sequence of instruction fetch (IF) instructions and at least one set of non-control instructions as presently claimed in the amended claim 12. Kitamura does not describe an instruction fetch (IF) memory storing the sequence of IF instructions. Kitamura also does not describe a programmable instruction fetch mechanism that is programmed by the IF instructions to fetch and execute IF instructions in a sequencing order, wherein the sequencing order is controlled by information contained in each of the IF instructions, as presently claimed in the amended claim 12. Kitamura also does not describe at least one non-control instruction memory (IMemory) storing the at least one set of non-control

instructions, whereby an IF instruction is formatted to identify at least one address of the at least one IMemory, as presently claimed in the amended claim 12.

Nothing in Kitamura teaches or makes obvious claims 1 or 12 as presently amended. These claims are not taught, are not inherent, and are not obvious in light of Kitamura. Since claims 2-4 and 6-11 depend from and contain all the limitations of the amended claim 1, claims 2-4 and 6-11 distinguish from Kitamura in the same manner as claim 1 and place claims 2-4 and 6-11 in order for allowance. Since claims 13-24 depend from and contain all the limitations of the amended claim 12, claims 13-24 distinguish from Kitamura in the same manner as claim 12 and place claims 13-24 in order for allowance.

Claims 1-9, 12, and 42 were rejected under 35 U.S.C. § 103(a) based on Hirata in view of Blackmon. Hirata discloses multiple instruction setup units which fetch and decode instructions where the decoded results are scheduled and sent to corresponding function units to be executed. Hirata, Abstract. Hirata also describes how an instruction fetch unit that is part of an instruction setup unit reads instructions by giving a memory or a cache the address designated by the program counter. The decode unit, also part of the instruction setup unit, decodes instructions and for those instructions which are to be executed, schedules them to be executed by function execution units. Hirata, col. 5, line 65 – col. 6, line 7. Hirata does not describe a method to transform a program by splitting the program into a set of control instructions and a set of arithmetic/logic instructions, reducing the set of AL instructions to a reduced set of AL instructions, assigning to each AL instruction an address from at least one AL memory, and generating instruction fetch (IF) instructions in a sequencing order specified by the IF

instructions for programmably selecting AL instructions to be fetched from said at least one AL memory, where the IF instructions have an IF instruction format which identifies at least one assigned address of at least one AL instruction as is presently claimed in amended claim 1. Hirata also does not describe a code splitting tool for transforming a program by generating an instruction addressing program as a sequence of instruction fetch (IF) instructions and at least one set of non-control instructions as presently claimed in amended claim 12. Hirata also does not describe a code splitting tool for transforming a program by generating an instruction addressing program as a sequence of instruction fetch (IF) instructions and at least one list of non-control instructions, where the IF instructions were not used in the program as is presently claimed in amended claim 42.

Nothing in Hirata teaches or makes obvious the limitations claimed in combination by claims 1, 12, or 42. The claims are not taught, are not inherent, and are not obvious in light of Hirata. Since claims 2-4 and 6-11 depend from and contain all the limitations of the amended claim 1, claims 2-4 and 6-11 distinguish from Hirata in the same manner as claim 1 and place claims 2-4 and 6-11 in order for allowance. Since claims 13-24 depend from and contain all the limitations of the amended claim 12, claims 13-24 distinguish from Hirata in the same manner as claim 12 and place claims 13-24 in order for allowance.

Blackmon describes an improved memory subsystem which determines the most efficient memory command to execute. Performance penalties, address dependencies, instruction history are example criteria that are examined to determine the most efficient memory command to execute. Blackmon, Abstract. The fetch command is defined as a command that requests data

from an address in memory. Blackmon, col. 7, lines 5 and 6. In contrast to Blackmon, the present invention claims in amended claim 12:

> A processor system comprising:
> a code splitting tool for transforming a program by generating an instruction addressing control program as a sequence of instruction fetch (IF) instructions and at least one set of non-control instructions;
> an instruction fetch (IF) memory storing the sequence of IF instructions;
> a programmable instruction fetch mechanism that is programmed by the IF instructions to fetch and execute IF instructions in a sequencing order, wherein the sequencing order is controlled by information contained in each of the IF instructions; and
> at least one non-control instruction memory (IMemory) storing the at least one set of non-control instructions, whereby an IF instruction is formatted to identify at least one address of the at least one IMemory and said programmable instruction fetch mechanism operates to fetch IF instructions from said IF memory and execute each fetched IF instruction to generate at least one IMemory instruction address to select at least one non-control instruction to be fetched from the at least one IMemory for execution.

The fetch commands of Blackmon are not the same as the IF instructions of the present invention which are of a different type and format than instructions used in the program and are used to identify at least one address of the at least one IMemory containing the at least one set of non-control instructions. Nothing in Blackmon teaches or makes obvious the limitations claimed in combination by claims 12 or 42.

Kishida describes an iterative start or a macro instruction which are used to set up a group of instructions to be sequentially fetched and executed. The iterative start and the macro instructions are stored in the program containing the group of instructions. Kishida, Figs. 6 and 12, col. 8, lines 34-38, and col. 13, lines 46-55. In contrast to Kishida, the present invention uses an IF instruction which is stored in an instruction fetch (IF) memory. The instructions to be sequentially fetched are stored in the at least one IMemory which is a different memory than the

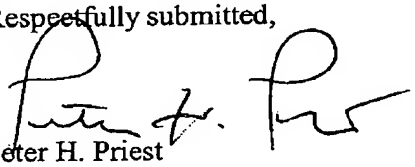
IF memory. The IMemory contains the at least one set of non-control instructions, as presently claimed in claim 12 and 13. Nothing in Kishida teaches or makes obvious claims 12-15.

Claim 25 has been cancelled without prejudice. The Official Action objected to claim 27 as being dependent upon a rejected base claim, but indicated that it would be allowable if rewritten in independent form. Claim 27 has been rewritten in independent form containing all the limitations of the base claim 25 from which it previously depended. Claims 26, 29-36, 40, and 41 have been amended to depend from claim 27. The grounds of rejection for claim 25 and its dependent claims are moot. Since claims 28, 37-39 and amended claims 26, 29-36, 40, and 41 depend from and contain all the limitations of the amended claim 27, claims 28, 37-39 and amended claims 26, 29-36, 40, and 41 are placed in order for allowance.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600